

This listing of claims will replace all prior versions, and listings, of claims in the application.

**Listing of Claims:**

1. (Currently Amended) A computer implemented method for continuing a preorder traversal of a binary tree formed by a family of nodes when a last node visited in a first partial traversal no longer exists in the tree, each node in the family comprising a child pointer, a sibling pointer, and a unique counter value, the method comprising:  
receiving as input a continuation node and a lineage for the continuation node, the continuation node comprising the last node visited in the first partial traversal and lineage comprising an output of a first partial preorder traversal of the binary tree;  
~~determining~~locating an updated continuation node ~~in the binary tree~~ by locating a first node currently existing in the tree included in the lineage of the continuation node having a level higher than any other nodes currently existing in the tree included in the lineage of the continuation node~~comparing the unique counter values of the family of nodes to a current node~~, the updated continuation node comprising the first currently existing node; and  
continuing the preorder traversal of the ~~binary~~ tree from the updated continuation node.
2. (Currently Amended) The method of claim 1 wherein the ~~determining~~comparing step follows the lineage of the continuation node.
3. (Original) The method of claim 1 wherein the binary tree comprises a dynamic binary tree.
4. (Original) The method of claim 1 wherein the binary tree represents a general tree.
5. (Original) The method of claim 1 wherein the binary tree represents a family of related processes.

6. (Original) The method of claim 1 wherein the binary tree represents a disk file directory structure.
7. (Original) The method of claim 1 wherein the binary tree represents a computer program structure.
8. (Original) The method of claim 1 wherein the nodes in the family further comprise a parent pointer.
9. (Currently Amended) The method of claim 1 wherein the lineage of the continuation node ~~an abbreviated continuation node lineage~~ is received as the input is abbreviated.
10. (Original) The method of claim 1 wherein the binary tree resides in a first environment and the input is received from a second distinct environment.
11. (Currently Amended) A computer-readable medium having ~~stored thereon~~ a data structure stored thereon, the data structure being used to for managing assist in the traversal of a tree comprised of a plurality of transitory nodes ~~elements related by hierarchy~~, the data structure representing a one of the nodes last visited during a partial traversal of the tree, elements and comprising:  
a data value field, a child pointer field, a sibling pointer field, ~~and~~ a unique counter field, and a lineage of the node.
12. (Original) The computer-readable medium of claim 11 wherein the data structure further comprises a parent pointer field.
13. (Canceled)
14. (Currently Amended) The computer-readable medium of claim 11 wherein the ~~elements comprise nodes in~~ tree comprises a binary tree.

15. (Original) The computer-readable medium of claim 11 wherein the unique counter field is populated with a non-decreasing counter value.
16. (Currently Amended) The computer-readable medium of claim 11 wherein the nodes~~elements~~ represent a process family structure.
17. (Currently Amended) The computer-readable medium of claim 11 wherein the nodes~~elements~~ represent a disk file directory structure.
18. (Currently Amended) The computer-readable medium of claim 11 wherein the nodes~~elements~~ represent a computer program structure.
19. (Currently Amended) A computer implemented method for locating an updated continuation node in a dynamic binary tree formed by a family of nodes when a last node visited in a first partial traversal of the tree no longer exists, each node in the family comprising a child pointer, a sibling pointer, and a unique counter value, the method comprising:
  - (a) receiving as input a continuation node and an abbreviated lineage for the continuation node, the continuation node comprising the last node visited in the first partial traversal of the tree; and
  - (b) traversing the nodes in the tree along the abbreviated lineage until the counter value indicates that a first valid node beyond the continuation node has been reached.
20. (Canceled)
21. (Currently Amended) The method of claim ~~19~~20 wherein step (b) comprises:
  - ~~(b)(1) determining whether the continuation node still exists in the tree;~~
  - (b)(~~12~~) if the result of step (b)(~~1~~) is no, determining whether the continuation node has a depth equal to or less than zero; and
  - (b)(~~23~~) if the result of step (b)(~~12~~) is no, traversing the nodes in the tree

along the abbreviated lineage until the counter value indicates that a first valid node beyond the continuation node has been reached.

22. (Original) The method of claim 19 wherein steps (a) and (b) are performed in a first data environment and the input is received from a second process in a second non-native data environment.
23. (Original) The method of claim 22 further comprising:
  - (c) passing as an output the updated continuation node to the second process.
24. (Original) The method of claim 19 wherein step (a) comprises:
  - (a)(1) receiving as input a continuation node and a lineage for the continuation node; and
  - (a)(2) extracting an abbreviated lineage for the continuation node from the continuation node lineage.
25. (Original) The method of claim 19 wherein the binary tree represents a general tree.
26. (Original) The method of claim 19 wherein the binary tree represents a family of related processes.
27. (Original) The method of claim 19 wherein the binary tree represents a disk file directory structure.
28. (Original) The method of claim 19 wherein the binary tree represents a computer program structure.

**DOCKET NO.:** TN226  
**Application No.:** 10/003,586  
**Office Action Dated:** 02/25/2005

**PATENT**

29. (Original) The method of claim 19 wherein the nodes in the family further comprise a parent pointer.
30. (New) The method of claim 1 wherein the determining step comprises comparing unique counter values of nodes currently existing in the tree with the continuation node unique counter value.